What is claimed is:

1. A processor comprising:

a plurality of logic components organized in a pipeline, wherein one of the logic components is an execute component;

a mechanism for indicating that an interrupt is pending;

logic capable of determining whether execution of a current instruction within the execute component will impact the execution of the pending interrupt; and

logic capable of performing a partial flush of the pipeline, wherein only instructions between the current instruction and a first instruction associated with the pending interrupt are flushed.

2. The processor of claim 1, wherein the plurality of logic components comprises a fetch component and a decode component, in addition to the execute component.

3. The processor of claim 1, wherein the plurality of logic components comprises an address generation component, a fetch component, an instruction queue, a decode component, a register file access component, a data access component, and a retirement component, in addition to the execute component.

4. The processor of claim 1, wherein the mechanism for indicating that the interrupt is pending comprises an electrical signal that is input to pipeline components that precede the execute component in the pipeline.

5. The processor of claim 1, wherein the execute component comprises logic for generating the electrical signal in response to an exception request.

6. The processor of claim 1, further comprising

a mechanism for indicating that a second interrupt is pending;

logic capable of determining whether execution of the current instruction within the execute component will impact the execution of the pending second interrupt; and

logic capable of performing a partial flush of the pipeline, wherein only instructions between the current instruction and a first instruction associated with the pending second interrupt are flushed.

7. The processor of claim 6, wherein the interrupt is a normal interrupt and the second interrupt is an fast interrupt.

8. The processor of claim 1, wherein the logic capable of determining whether execution of current instruction within the execute component will impact the execution of the pending interrupt is configured to determine whether the current instruction is one that will disable the pending interrupt.

9. The processor of claim 1, wherein the logic capable of determining whether execution of current instruction within the execute component will impact the execution of the pending interrupt is configured to determine whether the current instruction is a branch instruction.

10. The processor of claim 1, further including return instruction logic capable of setting a return address for instruction execution upon return from an interrupt service routine of the pending interrupt, the return instruction logic being operative in association with the logic capable of performing a partial flush.

11. The processor of claim 10, wherein the return instruction logic is configured to save, as a return address from the interrupt service routine, an address of the current instruction, if execution of the current instruction would result in a disabling of the pending interrupt.

12. The processor of claim 10, wherein the return instruction logic is configured to save, as a return address from the interrupt service routine, the correct destination address, if the current instruction is a mispredicted branch instruction.

13. A pipelined processor comprising:

interrupt logic configured to indicate that an interrupt is pending;

determining logic configured to determine whether execution of a current instruction within an execute stage of the pipelined processor will impact the execution of the pending interrupt; and

flush logic configured to flush only instructions between the current instruction and a first instruction associated with the pending interrupt.

14. The pipelined processor of claim 13, further comprising a mechanism for indicating that an interrupt is pending.

15. The pipelined processor of claim 14, wherein the mechanism is configured to generate an electrical signal in response to an exception request.

16. A portable electronic device containing the pipelined processor of claim 13.

17. A method executed by a processor while an interrupt is pending and before an interrupt service routine has begun execution, the method comprising:

determining whether execution of a current instruction will impact the execution of the pending interrupt; and

performing a partial flush of pipeline stages, wherein only instructions between the current instruction and a first instruction associated with the pending interrupt are flushed.

18. The method of claim 17, further comprising generating a signal indicating that an interrupt is pending.

19. The method of claim 18, wherein the generating a signal generates the signal in response to an exception request.

20. The method of claim 18, further comprising communicating the signal to all pipeline stages preceding an execute stage.

21. The method of claim 18, further comprising setting a counter, used to count pipeline stages, in association with the signal.

22. The method of claim 17, wherein the performing a partial flush of pipeline stages includes saving, as a return address from the interrupt service routine, an address of the current instruction, if execution of the current instruction would result in a disabling of the pending interrupt.

23. The method of claim 17, wherein the performing a partial flush of pipeline stages includes saving, as a return address from the interrupt service routine, a correct destination instruction address, if the current instruction is a mispredicted branch instruction.

24. A processor comprising:

a plurality of logic components organized in a pipeline, wherein one of the logic components is an execute component;

a mechanism for indicating that an exception is pending;

logic capable of determining whether execution of a current instruction within the execute component will impact the execution of the pending exception; and

logic capable of performing a partial flush of the pipeline, wherein only instructions between the current instruction and a first instruction associated with the pending exception are flushed.

25. The method of claim 24, wherein the exception is an interrupt.

26. A method executed by a processor while an exception is pending and before an exception service routine has begun execution, the method comprising:

determining whether execution of a current instruction will impact the execution of the pending exception; and

performing a partial flush of pipeline stages, wherein only instructions between the current instruction and a first instruction associated with the pending exception are flushed.